IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION FOR U.S. LETTERS PATENT

Title:

SYSTEM AND METHOD FOR PROCESSING TRAINING DATA FOR A STATISTICAL APPLICATION

Inventor:

Jonathan Qiang Li
600 Rainbow Drive, Apt. 149
Mountain View, California 94041
Citizenship: China

SYSTEM AND METHOD FOR PROCESSING TRAINING DATA FOR A STATISTICAL APPLICATION

TECHNICAL FIELD

[0001] Embodiments are directed to processing training data for statistical classification applications.

BACKGROUND

[0002] Statistical classification has two widely recognized meanings. First, based upon a set of observations or data, statistical classification seeks to establish the existence of classes or clusters in the data. This type of statistical classification is referred to as unsupervised learning (or clustering). Secondly, the existence of classes may be known beforehand. In this second case, statistical classification seeks to establish a rule or rules whereby a new observation is classified into one of the known existing classes. This type of statistical classification is known as supervised learning.

[0003] Supervised learning possesses wide applicability to industrial and technical applications. For example, supervised learning may be used to establish a rule or rules for machine vision recognition. The machine vision recognition based upon the established rule(s) may be used to guide or control an automated fabrication process.

[0004] In supervised learning, a set of measurements are selected that are believed to be indicative of the defined classification(s). Training data is created based upon the selected measurements. Each element in the training data is labeled according to the defined classifications. Upon the basis of the label training data, various methodologies may be used to classify subsequently observed data elements.

[0005] The "nearest neighbor" classification methodology measures the distance (e.g., calculated using a suitable weighted metric) from an observed data element to each data element in the training data. The N-closest data elements from the training data are selected. The most frequently occurring class in the N-closest data elements is used to classify the observed data element.

[0006] The classification methodology assumes that the classifications of the training data elements are correct. However, the classifications can possess a number of errors for a variety of reasons. The amount of misclassification is related to accuracy of the classification methodology. Specifically, the greater amount of misclassification in the training data leads to reduced accuracy of the classification performance. Thus, data integrity of the classification data is an important consideration in supervised learning applications.

SUMMARY

[0007] Representative embodiments are directed to systems and methods for processing training data for a supervised learning application. In one embodiment, confidence values are calculated for training data elements to identify the probabilities of the training data elements belonging to identified classes. An interactive scatter plot is generated using the calculated confidence values. The scatter plot visually indicates the confidence values of points in the scatter plot. Accordingly, the user is able to identify potentially misclassified training data elements. The user may select training data elements from the scatter plot. In one embodiment, an image file of an object associated with the training data element is displayed upon selection of a corresponding point from the scatter plot. The user may reclassify the training data element. Alternatively, the user may delete the training data element. Upon reclassification, probabilistic models corresponding to the various classes are revised. Using the revised models, the confidence values may be recalculated and the scatter plot displayed using the recalculated confidence values.

BRIEF DESCRIPTION OF THE DRAWINGS

[0008] FIGURE 1 depicts an interactive scatter plot according to one representative embodiment.

[0009] FIGURE 2 depicts an image window that may be presented to a user in response to selection of a point of a scatter plot according to one representative embodiment.

[0010] FIGURE 3 depicts a flowchart for processing training data using an interactive scatter plot according to one representative embodiment.

[0011] FIGURE 4 depicts a computer system that implements an interactive scatter plot utility according to one representative embodiment.

DETAILED DESCRIPTION

[0012] Referring now to the drawings, FIGURE 1 depicts interactive scatter plot 100 that enables training data for a supervised learning application to be processed according to one representative embodiment. Interactive scatter plot 100 is a graphical user interface (GUI) that includes a plurality of points. Each of the points corresponds to an underlying training data structure. Each training data structure may include member variables that correspond to the features of the classification scheme (shown as FEATURES 1 and 2 in FIGURE 1). Each training data structure may also include a member variable that identifies one of a plurality of classes (shown as classes 101 and 102 in FIGURE 1) to which the training data structure currently belongs.

[0013] The points in interactive scatter plot 100 visually indicate the probability that the corresponding training data structures are correctly classified. For example, point 103 is shown to possess a probability of 0.13 of belonging to class 101. Because of the visual presentation of interactive scatter plot 100, a user can efficiently identify point 103 and examine the training data structure in greater detail to determine whether it is, in fact, misclassified. Other methods may be used to visually indicate the probabilities of correct classifications. For example, points possessing a probability less than a threshold value may be displayed using a predefined color.

[0014] In one representative embodiment, image window 200 as shown in FIGURE 2 may be presented upon selection of a point from scatter plot 100 as an example. Image window 200 may depict the object from which the feature values were derived. The values of the feature elements can be presented in association with the image of the object if desired. Also, the current classification may be noted. The classification may be included within graphic user interface control 201 to enable the user to modify the classification upon visual inspection of the object. If a training data structure is manually verified or reclassified, a suitable indication may be noted within scatter plot 100 (e.g., changing the representation of the point from a "circle" to another shape). Image window 200 is by way of example only. Any suitable user interface may be employed to enable a user to verify and/or modify training data structures according to representative embodiments.

[0015] To calculate the probabilities for interactive scatter plot 100, probabilistic models are specified. The probabilistic models summarize the location and shape of the data distribution for the respective classes. By employing such models, potential errors in the training data can be identified. A number of statistical models may be employed for this purpose. The choice of a model may depend upon a number of factors such as accuracy, ease of use, speed, sample sizes, and/or the like. For example, "nonparametric models" (such as the Kernel Density Estimator) can be used to define the location and shape of a data distribution in a relatively general manner. A Gaussian mixture model can be employed (i.e., a distribution that possesses multiple Gaussian components) for distributions possessing multiple modes. Alternatively, a Gaussian distribution can be employed if a relatively low complexity model is selected.

[0016] The notation P(x|θ) may be used for the probabilistic model where x is the data value vector and θ is the model parameter vector. In a Gaussian model,

$$P(x|\theta) = P(x \mid \mu, \Sigma) = \frac{1}{\sqrt{(2\pi)^k \mid \Sigma \mid}} \exp\{-\frac{1}{2}(x - \mu)^T \Sigma^{-1} (x - \mu)\},$$

where k is the dimension of the data (x), μ is the mean vector of length k, and Σ is the covariance matrix of size k by k. In a two-class classification problem, two models are specified (one for each class).

[0017] After the selection of the model(s), the model parameters are estimated from the training data elements: $X_1, \ldots\ldots X_n$. In general, the training data elements X are vectors of length k: $X_i = (x_{i1}, \ldots\ldots x_{ik})^T$. In one embodiment, the model parameters are estimated from the training data elements using the Maximum Likelihood Estimation (MLE) method. For the Gaussian model, the MLE method provides reasonable estimators. The estimator for the mean is the sample mean, i.e., $\mu = (\mu_1, \ldots, \mu_k)^T$, where $\mu_p = \frac{1}{n} \sum_{i=1}^{n} x_{ip}$, $p = 1, 2, \ldots k$ and n is the number of training elements associated with the particular class. The estimator for the covariance matrix is given by:

$$\Sigma = \begin{pmatrix} \sigma_{11} & \sigma_{12} & \cdots & \sigma_{1k} \\ \sigma_{21} & \sigma_{22} & \cdots & \sigma_{2k} \\ \cdots & & & \\ \sigma_{k1} & \sigma_{k2} & \cdots & \sigma_{kk} \end{pmatrix},$$

where the elements of the matrix are estimated by

$$\sigma_{pq} = \frac{1}{n-1}\sum_{i=1}^{n}(x_{ip} - \mu_p)(x_{iq} - \mu_q), p = 1,2,...k, q = 1,2,...,k.$$

[0018] After generating the estimators for each class, the confidence values can be computed using Bayes theorem. In this context, confidence refers to the probability of a data point belonging to its identified class. Bayes theorem provides a systematic mechanism to update such probabilities with each new training data element and/or with each reclassification of an existing training data element.

[0019] Before training data elements and, hence, measurements are available, the prior probability of a data point being within each class is 0.5. In other words, it is equally likely that a point belongs to either class. At this stage, the following notation may be employed l_1=P(class1)=0.5, l_2=P(class2)=1-l_1=0.5. Once a measurement is made (a training data element is defined), the probabilistic models for the classes can be specified or revised by estimating the parameters of the models using the respective training data element. Let P(X|class1)= P(X|θ_1), P(X|class2)= P(X|θ_2), where X represents the respective training data element, θ_1 refers to the model estimators of the first class, and θ_2 refers to the model estimators of the second class.

[0020] Bayes theorem relates the conditional probability of one event given another event with the joint probability of the two events as follows:

$$l_1^{update} = P(class1 \mid X) = \frac{P(X \mid class1)P(class1)}{P(X)} = \frac{P(X \mid class1)P(class1)}{P(X \mid class1)P(class1) + P(X \mid class2)P(class2)},$$

$$= \frac{P_1(X \mid \theta_1)l_1}{P_1(X \mid \theta_1)l_1 + P_2(X \mid \theta_2)l_2}, \text{ and}$$

$$l_2^{update} = 1 - l_1^{update}.$$

[0021] Thus, an iterative methodology may be employed that refines the class probabilities based upon the prior stage class probabilities. Accordingly, the iterative methodology enables the confidence values to be refined by the addition or modification of training data elements. Also, Bayes theorem can be applied to both univariate or multivariate models. A multivariate model refers to a model in which multiple feature elements are employed to characterize training data elements. If a multivariate model is employed, the features may be assumed to be statistically independent.

[0022] FIGURE 3 depicts a flowchart for processing training data structures for a statistical learning application using an interactive scatter plot according to one representative embodiment. The process flow of FIGURE 3 may be implemented as software instructions executed by a suitable processor system.

[0023] In step 301 of FIGURE 3, the training data structures are retrieved from memory (e.g., a suitable storage peripheral). Each training data structure may include one or several feature variables or members. The feature members may store the values associated with the measurements. Each training data structure may include a classification variable or member to identify the class to which the training data structure has been assigned.

[0024] In step 302, the probabilities of the multiple classes are iteratively calculated. In step 303, using the class probabilities and the model estimators, a confidence value is calculated for each training data structure. The confidence value is the probability that the respective training data structure belongs to its identified class. As previously noted, a variety of probabilistic models can be employed depending upon the particular application. Additionally, the probabilistic models can be univariate or multivariate.

[0025] In step 304, a scatter plot is created using the feature variables of the training data structures. In one embodiment, different shapes may be used for points depending upon whether the points have been previously verified or reclassified by the user. In step 305, points in the scatter plot may be annotated with the calculated confidence values. In step 306, points in the scatter plot may be changed according to confidence values falling below a threshold value. The threshold value may be manually set by the user if desired. In step 307, user input is received to select a point from the scatter plot.

[0026] In step 308, the training data structure associated with the selected point is retrieved. In step 309, the image file associated with the respective training data structure is retrieved. In step 310, an image window is displayed. In step 311, additional user input is received to verify the current classification, modify the classification, or delete the training data structure. In step 312, a logical comparison is made based upon the user input. If the user verified the current classification, the process flow returns to step 304. If the user selected the delete option, the process flow proceeds to step 313 where the delete operation is performed. If the user selected the modify option, the process flow proceeds to step 314 where the classification member is changed according to the user input.

[0027] From either step 313 or step 314, the process flow returns to step 302 to revise the class probabilities based upon the deletion or modification. Steps 303-306 are also repeated thereby presenting a revised scatter plot to the user. Specifically, because the training data was corrupted by a suspect classification, the revision enables the model estimates to be made more accurate. The improved accuracy of the model estimates also enables the confidence values to be calculated more accurately. Thus, it is seen that an iterative process occurs that enables the user to efficiently verify the classifications of a set of training data.

[0028] FIGURE 4 depicts computer system 400 that may be used to execute interactive scatter plot utility 406 according to one representative embodiment. Computer system 400 includes typical processing resources such as processor 401, display 402, input peripheral(s) 403, and non-volatile memory 405. Non-volatile memory 405 (which may be implemented using any suitable computer readable medium) stores the executable software code defining interactive scatter plot utility 406. Interactive scatter plot utility 406 may be implemented using the flowchart described with respect to FIGURE 3 as an example. Interactive scatter plot utility 406 may be organized according to a number of routines, functions, code segments, and/or the like. As shown in FIGURE 4, interactive scatter plot 406 includes class probability calculation routine 407, confidence value calculation routine 408, scatter plot display routine 409, user input routine 410, and image window display routine 411. Interactive scatter plot utility 406 may access various data stored within non-volatile memory 405 such as training data structures 412 and image files 413.

[0029] Representative embodiments enable a user to efficiently process training data. By visually identifying training data elements that possess relatively low confidence values, the user's attention may be brought to suspect training data in a relatively quick manner. Moreover, by enabling the user to access the underlying training data and/or image files, the user can verify, reclassify, or delete training data elements as appropriate. In response to such user input, confidence values may be recalculated and the scatter plot can be redisplayed using the more accurately calculated confidence values.